09/243,108 **

L3: Entry 24 of 97 File: USPT Nov 19, 2002

DOCUMENT-IDENTIFIER: US 6481632 B2
TITLE: Delegated management of smart card applications

Abstract Text (1):
A smart card architecture includes a run-time environment, a card manager, one or more security domains, a provider application and an issuer application. One or more APIs provide communication. The life cycle of the card and card manager includes states: Pre-production, Ready, Initialized, Secured, Locked and Terminated. The life cycle of an application includes states: Installed, Selectable, Personalized, Blocked, Locked and Deleted. A card registry keeps track of card manager and application data elements. The functionality of a security domain on a smart card is extended to allow it to perform delegated management of smart card applications: delegated loading, installation and/or deletion of an application. A provider of an application is assured of more direct control and management of their application, yet an issuer still maintains some control over the management of the card. The card issuer empowers application providers to initiate changes to the issuer's smart cards that are pre-approved by the card issuer. A method of delegated loading of an application onto a smart card first receives a load command from an application provider via a card acceptance device. The load command includes an indication of an application to be loaded and an appended command authentication pattern. Next, the load command is verified using the command authentication pattern. Then, an application is received from an application provider via the card acceptance device; the application also includes an appended application authentication pattern which is used to verify the application. Finally, the application is loaded into memory of the smart card.

Brief Summary Text (2):
The present invention relates generally to smart cards. More specifically, the present invention relates to a technique for delegating the management of applications on a smart card such as loading, installation and deletion.

Brief Summary Text (4):
Smart card technologies hold great promise as the replacement for magnetic stripe card technology. The adoption of smart cards, however, on a massive scale has been slow to develop. One reason for this slow adoption is the lack of standards among the many different vendor implementations of smart cards and the difficulties with implementing a new technology.

Brief Summary Text (5):
Recently, significant standards in the smart card area have been created. The standards, however, have been primarily targeted at either low levels of interoperability, such as the mechanical and electrical standards specified in the EMV specifications, or at the application layer in terms of developing standard chip credit, debit and purse applications. The main benefit of the standards has been realized in single-application smart cards, but has not significantly improved the situation for multi-applications smart cards.

Brief Summary Text (6):
The mid-1990s saw the introduction of various open systems standards for application development. For example, three technologies in this area are JAVA Card from Sun Microsystems, Inc., Smart Card for Windows from Microsoft Corporation, and MULTOS from MAOSCO, Ltd. These technology standards provide an important part of the

solution toward common programming standards allowing application portability between different manufacturers card implementations. Other recent efforts have also addressed particular issues with multi-application smart cards. For example, U.S. patent application Ser. No. 09/046,994 filed Mar. 24, 1998, and U.S. patent application Ser. No. 09/046,993 filed Mar. 4, 1998 address issues related to post-issuance downloading and life cycle, each of which are hereby incorporated by reference.

Brief Summary Text (7):
In prior art smart cards only the issuer of the card has been allowed to perform certain management functions of applications such as loading an application onto the card, installing the application and deleting the application from the card. This reliance upon the issuer exclusively for loading, installing and deleting applications can lead to some difficulties. For example, should a store develop a loyalty application for its customers that it wishes to load and install onto their customer's smart cards, the store would be precluded from doing so if only the card issuer is allowed to perform such functions. Arranging for the store to contact the issuer, and arranging for the issuer to load the store's application onto its customer's smart cards presents basic logistical problems such as application security and access to cards. For example, it would be best if the store could download an application onto a customer card while the customer visited the store. If only the issuer can download the application, it becomes more difficult to access the customer card.

Brief Summary Text (9):
Other mechanical difficulties are presented should a customer desire to download and install a third party's application at the third party's site if only the issuer is allowed to download and install an application. For example, should a customer wish to download a loyalty application while at the third party's place of business during a smart card transaction, it would be first be necessary for the card acceptance device to connect to the issuer host to download the loyalty application, and then connect to the third party's host computer in order to receive custom information for the initialization and personalization of that application. Such multiple connections at load time make the transaction more complex, time consuming and are more prone to failure. In addition, as a practical matter, should a customer wish to download and install an application onto his smart card, it is more than likely that the customer is physically present at a third party site rather than at the issuer's site.

Brief Summary Text (14):
This concept of delegated management allows the card issuer the option of empowering application providers to initiate changes to the issuer's smart cards that are pre-approved by the card issuer. This pre-approval ensures that only card content changes that the card issuer has approved will be accepted and processed by a card manager of a smart card. This delegation of control in the card update process allows application providers more flexibility in managing their own applications on the card issuer's cards.

Brief Summary Text (16):
In another embodiment, a system for delegated loading of an application onto a smart card includes a host computer under control of an application provider and a software application to be loaded onto a smart card. The application includes an appended application authentication pattern produced by an issuer of the smart card that verifies the application to the smart card. The system also includes a smart card acceptance device linked to the host computer and a smart card included in the card acceptance device. The smart card includes code arranged to verify the application using the application authentication pattern. Thus, the application provider is allowed to load the application onto the smart card.

Drawing Description Text (11):
FIG. 7C is a flow diagram describing one embodiment of the download application step of FIG. 7A.

Drawing Description Text (15):
FIG. 11 illustrates an embodiment in which an application may be downloaded from an

application provider host computer to a smart card in a delegated manner.

Detailed Description Text (2):
The present invention is suitable for use with either single or multi-application smart cards. A multi-application smart card may come from a variety of manufacturers and may use any of a number of operating systems. By way of example, a smart card may use the JAVA Card operating system or the Smart Card for Windows operating system. As used herein, "smart card" refers to any of these single-application or multi-application smart cards. In one particular embodiment, the present invention works well with the "Open Platform" architecture as defined in Open Platform Card Specification Version 2.0, Apr. 19, 1999, available from Visa International Service Association. This architecture in one embodiment is based upon the JAVA Card operating system and provides a hardware-neutral, vendor-neutral, application-independent card management standard. The standard provides a common security and card management architecture and defines a flexible and powerful standard for card issuers to create multi-application smart cards.

Detailed Description Text (3):
Smart Cards

Detailed Description Text (4):
The present invention is applicable to smart cards. Also termed chip cards, integrated circuit cards, memory cards or processor cards, a smart card is typically a credit card-sized plastic card that includes one or more semiconductor integrated circuits. A smart card can interface with a point-of-sale terminal, an ATM, or with a card reader integrated with a computer, telephone, vending machine, or a variety of other devices. The smart card may be programmed with various types of functionality such as a stored-value application, a credit or debit application, a loyalty application, cardholder information, etc. Although a plastic card is currently the medium of choice for smart cards, it is contemplated that a smart card may also be implemented in a smaller form factor, for example, it may attach to a key chain or be as small as a chip module. A smart card may also be implemented as part of a personal digital assistant, telephone (such as a subscriber identification module), or take a different form. The below description provides an example of the possible elements of a smart card, although the present invention is applicable to a wide range of types of smart cards.

Detailed Description Text (10):
FIG. 1 illustrates symbolically an environment in which Open Platform architecture 10 provides benefits for smart card holders, issuers, application developers and other entities. Although the present invention works well within architecture 10, it is also suitable for use in other architectures. Open Platform architecture 10 embodied within a smart card 20 provides card issuers an architecture for managing smart cards. Architecture 10 gives card issuers the power to manage and change the content of their cards while also offering them the flexibility to share control of their cards with other business entities. Preferable, ultimate control rests with the card issuer, but through use of architecture 10 other business entities are allowed to manage their own applications on the card issuers cards as appropriate. An issuer personalizes new cards received from a card supplier and then issues these cards to customers. Personalization may also be performed by the card supplier or by a personalization bureau. An issuer may be any suitable issuing entity such as a bank, financial institution, telecommunications network operator, a service association, a merchant or other organization, or even an agent acting for an issuer.

Detailed Description Text (11):
As shown symbolically, in FIG. 1 a wide variety of systems and devices benefit through use of architecture 10. A smart card 20 has been previously described and its relationship with architecture 10 will be further explained below. A card acceptance device 22 (also termed card reader or terminal) may contain an application that interacts with architecture 10 within smart card 20, and can also download an application onto the smart card. A terminal management system 24 manages terminals and their respective applications. An application server 26 provides an application for a smart card or card reader. A personalization system 28 personalizes smart card applications. A card management system 30 manages an

issuer's card base and respective applications. A key management system 32 provides support for pre-issuance and post-issuance support for key generation and/or key storage and/or key retrieval. Application development tools 34 are used to develop smart card applications.

Detailed Description Text (12):
FIG. 2 illustrates in further detail Open Platform architecture 10 as it may be implemented upon a smart card (not shown). Run-time environment 102 includes both the actual hardware of the smart card as well as the operating system of the smart card. Architecture 10 may be implemented on top of any card run-time environment. Run-time environment 102 is responsible for providing a hardware independent application programming interface (API) for provider applications as well as a secure storable and executing space for applications, thus ensuring that each application's code and data are able to remain separate and secure from others. Architecture 10 may be used with any of a wide variety of physical smart cards available from a wide variety of manufacturers. Further, architecture 10 may be implemented on top of any suitable smart card operating system, such as JAVA Card and Smart Card for Windows, among others.

Detailed Description Text (33):
The Installed State 222 means the application executable has been properly linked and any necessary memory allocation has taken place so that the application may execute. The installation process does not include establishing the application as an externally visible application (the Selectable State), also, installation is not intended to personalize the application. Preferably, card manager 104 sets the life cycle of an application to the state Installed during the application installation process.

Detailed Description Text (42):
Application identifier 260 is a unique identifier for each application on the card and is used by the card manager for application selection. Application life cycle state 262 contains the current life cycle state of the application or security domain. Resource allocation 264 is a data element that contains a value for the total amount of resources that are available to an application. It is an application specific value and is used as a control mechanism by the card manager to limit the amount of resources that an application uses during run time. When additional resources are requested by an application. The card manager compares against this data element. Application privileges 266 are a set of data elements that indicate privileges for each application. A variety of privileges may be indicated for an application: the application is a security domain without delegated management privilege; the application is a security domain with data authentication pattern privileges; the application is a security domain with delegated management privilege; the application has card manager locking privilege; the application has card termination privilege; the application is the default selected application; and the application has privilege to change a card global PIN. Each privilege may be marked as true or false, and an application may have more then one privilege marked as true. The card manager may apply a set of rules to these privileges for management of the card in any suitable fashion.

Detailed Description Text (46):
The Open Platform architecture allows parties other than the card issuer such as application providers to load, install, and delete their own applications. In general, these processes are referred to as delegated management. In general it is desirable that a card issuer have complete control over the smart cards it issues. The card issuer, however, may not necessarily wish to manage all card content changes, especially when the content does not belong to the card issuer but to an application provider. This concept of delegated management is incorporated into the Open Platform architecture to allow the card issuer the option of empowering application providers to initiate changes to the issuer's smart cards that are pre-approved by the card issuer.

Detailed Description Text (48):
Delegated loading allows the application provider to establish a loading session for transferring their application files directly to their own security domains. Once each APDU command has been securely transferred onto the card, the security domain

passes it to the card manager for loading into persistent memory. The card manager is able to identify the authenticity of these processes through the use of a data authentication pattern applied to the install commands and the load file itself. The card manager does not verify the data authentication patterns applied to individual Load commands. The command related DAPs which the card manager does check are referred to as the Load and Install tokens.

Detailed Description Text (59):
In step 326 a cardholder inserts the smart card into any suitable card acceptance device. This insertion may be part of a regular transaction for the cardholder or it may be a special transaction solely for the purpose of downloading the new provider application. In step 330 the provider downloads the new application onto the smart card in the card acceptance device; this step will be described in further detail in FIG. 7C. In this fashion, the loading of the application has been delegated to an application provider.

Detailed Description Text (61):
FIG. 7B is a flow diagram that describes how an issuer approves an application for delegated loading and installation. Once an application provider has written an application for a smart card and desires to load that application onto an issuer's smart cards via the delegated loading process of the present invention, it provides the application to the issuer for approval. It should also be noted that the provider may also give the application to a trusted third party for approval. In step 340 the issuer performs testing of the application given to it by the application provider. Testing of an application for a smart card may be performed in any of a variety of ways and is a step understood in the art, and generally involves functional tests (optional) and security tests (mandatory). Testing of the application involves checking its operational behavior on a smart card, checking its operational memory requirements, etc., ensuring that the application is secure, and checking for viruses and card related threats. Once the issuer (or trusted third party) has tested the application and it to ensure that it behaves correctly, the application is "certified" and the issuer is ready to prepare the application for a delegated load and installation by the provider.

Detailed Description Text (72):
FIG. 7C is a flow diagram describing one embodiment of the download application step of FIG. 7A. In step 360 a data link is established between host computer 602 and smart card 604 while the card is in a card acceptance device. Preferably, as part of this procedure, a mutual authentication process is performed between the card and the host using a key set provided to the application provider by the issuer or other trusted third party (as previously described). Preferably, the security domain keys provided to the application provider are used so as to assure security domain 106 that the incoming information is coming from an authorized source are only seen by the application provider. In step 362, host 602 sends load command 500 to security domain 106 using APDU interface 610. In essence, this is a request for a load of an application.

Detailed Description Text (76):
Accordingly, in step 374 the host sends the application to security domain 106 over links 620. Preferably, included with the application is the data authentication pattern previously created for it by the issuer. In a preferred embodiment, the application is embedded within a load file such as is illustrated in FIG. 10 which itself is part of an APDU Load command. Of course, the application may be embodied in other types of commands, need not necessarily be part of a load file, or simply may be transmitted by itself along with it data authentication pattern. As with the load command, in step 376 the security domain passes the application on to the card manager. In step 378 the card manager authenticates the application by verifying its data authentication pattern that was created by the issuer previously. As mentioned above, the card manager may authenticate the data authentication pattern of the application using any of a variety of cryptographic techniques. If any authentication fails, the original memory contents are restored.

Detailed Description Text (77):
In step 382 the installer loads the actual application code into memory of the smart card and performs linking to any run-time libraries and other necessary steps. As

mentioned above, in a preferred embodiment, the installer performs a load by processing one or more APDU Load commands that contain the application. Assuming that loading and linking was performed successfully, in step 384 a confirmation message is sent from the installer to provider host computer 602 via the card manager and the security domain. Once host 602 has received the confirmation, it is notified that the application has been loaded successfully.

Detailed Description Text (97):
FIG. 11 illustrates an embodiment in which an application may be downloaded from an application provider host computer 602 to a smart card 604 in a delegated manner. Host computer 602 is any suitable computing device under control of an application provider that includes the load command 500, load file 560 and install command 520 that have been approved and received from the issuer.

Detailed Description Text (99):
In the prior art, an issuer by virtue of its secret keys would be able to talk directly to card manager 104 through APDU interface 612 to provide an application to be loaded onto the card. Installer 614 would accept this application via the card manager and install the application in the memory of the smart card. For security, the keys to access card manager 104 would not be accessible to parties other then the issuer, meaning that only an issuer could download an application. Through use of the present invention, a third party application provider is able to perform a delegated load of an application via security domain 106. Using keys previously received under an arrangement with an issuer, host computer 602 establishes a secure communication channel 620 to security domain 106 of smart card 104 in any suitable card acceptance device (not shown).

Detailed Description Text (100):
Security domain 106 then manages the downloading of load command 500, load file 560 and install command 520 onto smart card 604. In this fashion, these commands and the load file may then be delivered via an internal link 622 to card manager 104 using a delegated management interface 616. The card manager then passes the commands and load file to installer 614 for loading and installing an application onto a smart card. Installer 614 receives and process these commands from security domain 106 in much the same way as if these command had been received from an issuer via card manager 104. Further, the data authentication pattern present in the commands and in the application may be checked by the card manager to ensure the authenticity and integrity of the information as established by the issuer. Further details on loading and installation are provided in FIG. 7C.

Other Reference Publication (4):
Carol H. Fancher, "Smart Cards as Potential Applications Grow, Computers in the Wallet are Making Unobstrusive Inroads", Aug. 1996, Scientific American Website.

Other Reference Publication (5):
Jerome Svigals, "Smart Cards The New Bank Cards", 1985, MacMillan Publishing Company.

Other Reference Publication (8):
Hawkes et al., "Integrated Circuit Cards, Tags and Tokens", 1990, BSP Professional Books.

Other Reference Publication (9):
David Naccache, "Cryptographic Smart Cards", Jun. 3, 1996, IEEE Micro 1996 Website.

Other Reference Publication (10):
Zoreda et al., "Smart Cards", 1994, Artech House.

Other Reference Publication (35):
Hiro Shogase, "The Very Smart Card: A Plastic Pocket Bank", IEEE Sepctrum, Oct. 1988.

CLAIMS:

5. A system for delegated loading of an application onto a smart card, said system

comprising: a host computer under control of an application provider; a software application included in said host computer to be loaded onto a smart card, said application including an appended application authentication pattern produced by an issuer of said smart card that verifies said application to said smart card; a smart card acceptance device linked to said host computer; and a smart card included in said card acceptance device, said smart card including code arranged to verify said application using said application authentication pattern, whereby said application provider is allowed to load said application onto said smart card.

11. The system as recited in claim 5 further comprising: a network connection linked to the smart card issuer; computer readable code for sending the application from application provider to the smart card issuer; computer readable code for receiving the approved application and an appended application authentication pattern from the smart card issuer; and a storage device for storing the application and the appended application authentication pattern.